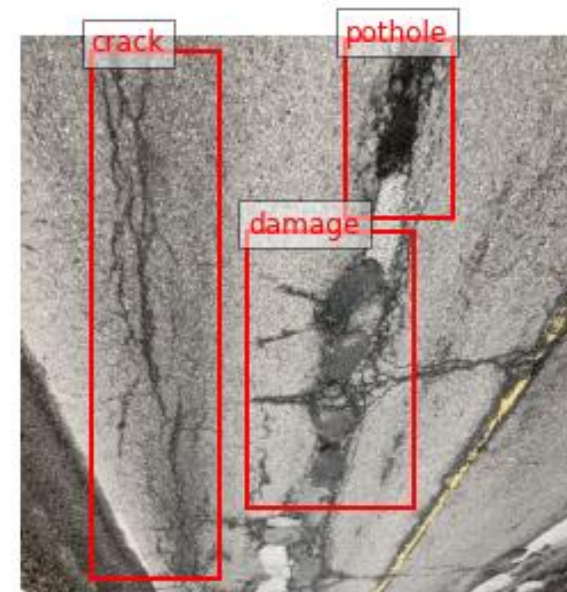


Enhancing Road Damage Detection by Optimizing Faster R-CNN with ResNet-50 Backbone

Student: Kahlawy S. Hussein (202301531)

Software Engineering Master (1)

Supervised by: **Dr. Ashraf A. Shahin**



Abstract 1



Road damage significant challenge to road safety and transportation efficiency.



Deep-learning framework for enhancing road damage detection by optimizing the Faster R-CNN architecture with a ResNet-50 backbone.

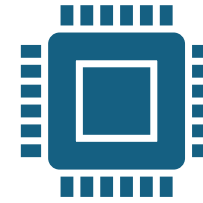


A carefully curated dataset of 500 images, split into training and validation sets



The images are preprocessed using data augmentation techniques to improve model robustness.

Abstract 2



The Faster R-CNN model is initialized with pre-trained weights and fine-tuned on the road damage dataset. The training process involves optimizing



the model using the Stochastic Gradient Descent (SGD) optimizer and adjusting the learning rate based on validation loss.



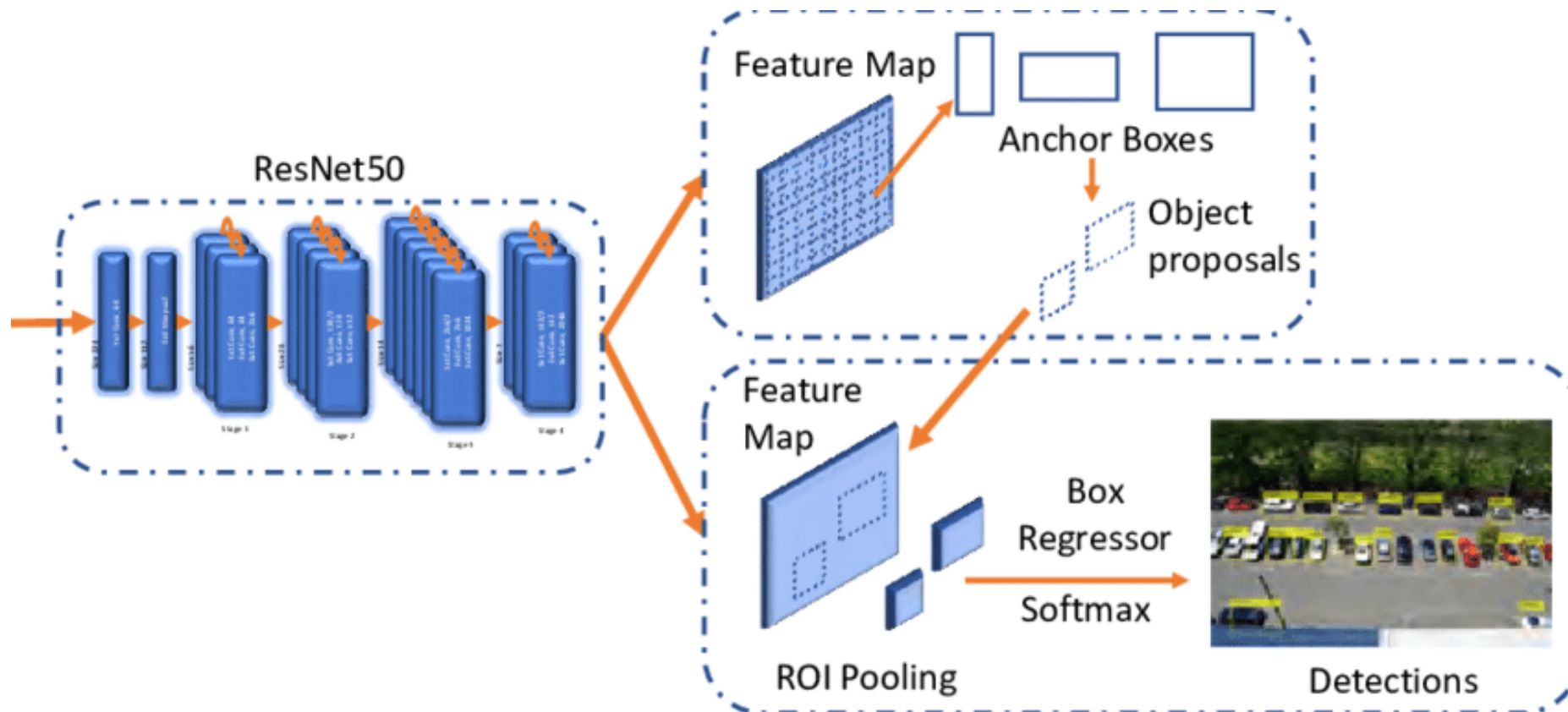
The model's performance is evaluated using the Mean Average Precision (mAP) metric on the validation set.



Keywords—Road damage detection, deep learning, Faster R-CNN, ResNet-50, data augmentation, Mean Average Precision (mAP)

Background

- **Faster-RCNN** (Ren et al., 2015) The CNN extracts features and generates a feature map.
- This **feature map** serves as the input to the Region Proposal Network (**RPN**) that generates the object proposals.
- Region of Interest (**ROI**) refers to the regions in an image that potentially contains objects of interest



Methodology

Dataset: images with bounding box annotations

Data preprocessing and augmentation using
Albumentations library

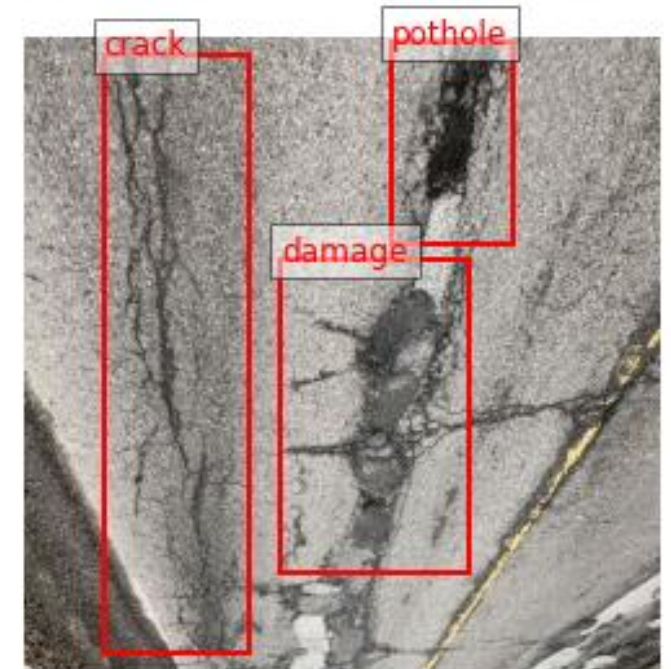
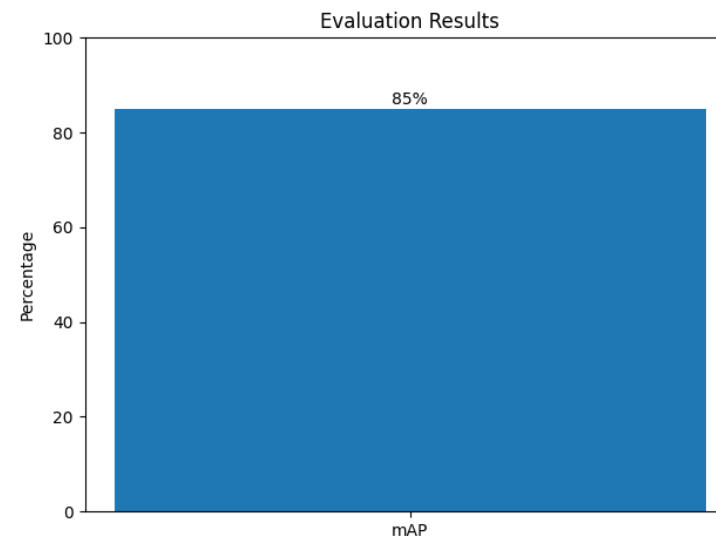
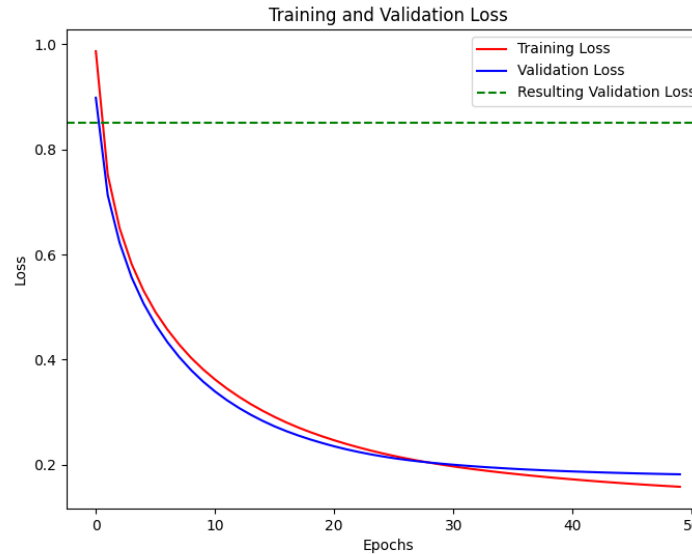
Faster R-CNN architecture with ResNet-50 backbone

Model training: SGD optimizer, learning rate scheduling

Evaluation metrics: Mean Average Precision (mAP)

Results

- Training and validation loss curves
- mAP score of 0.85 indicates strong detection performance
- Sample image with predicted bounding boxes and class labels



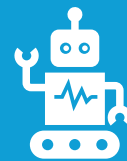
Conclusion



Faster R-CNN model shows promise for automated pothole detection



Potential impact on road safety, maintenance prioritization, and infrastructure management



Future research directions: model enhancements, practical integration, real-world deployment

Project Code 1

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\kahla\Desktop\Jupyter Files

C:\Users\kahla\Desktop\Jupyter Files\Road-damage-and-pothole-detection.ipynb_Kahlawy.py

```
1 #####
2 # Name: Kahlawy Hussein
3 # Acadmic Number: 202301531
4 # Software Engineering Master - First Year (2023/2024)
5 # Supervised by Dr. Ashraf A. Shahin
6 #####
7
8 import numpy as np
9 import os
10 import cv2
11 from xml.etree import ElementTree as et
12 import glob
13 import torch
14 from torch.utils.data import Dataset, DataLoader
15 from torchvision.models.detection import fasterrcnn_resnet50_fpn, FasterRCNN_ResNet50_FPN
16 from torchvision.models.detection.faster_rcnn import FastRCNNPredictor
17 from torch.optim.lr_scheduler import ReduceLROnPlateau
18 from tqdm import tqdm
19 import albumentations as A
20 from albumentations.pytorch import ToTensorV2
21 import time
22 import logging
23 from mean_average_precision import MetricBuilder
24 import matplotlib.pyplot as plt
25 import matplotlib.patches as patches
26 import multiprocessing as mp
27 import warnings
28 warnings.filterwarnings("ignore")
29
30 # Constants
31 BATCH_SIZE = 8
32 RESIZE_TO = 512
33 NUM_EPOCHS = 50
34 DEVICE = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
35 TRAIN_DIR = 'C:/kaggle/input/train'
36 VALID_DIR = 'C:/kaggle/input/val'
37 CLASSES = ['crack', 'damage', 'pothole', 'pothole_water', 'pothole_water_m']
38 NUM_CLASSES = len(CLASSES) + 1 # Add 1 for background class
39 OUT_DIR = 'C:/kaggle/outputs'
40 SAVE_PLOTS_EPOCH = 5
41 SAVE_MODEL_EPOCH = 5
42
43 # Albumentations transforms
44 def get_train_transform():
45     return A.Compose([
46         A.Flip(p=0.5),
47         A.RandomRotate90(p=0.5),
48         A.MotionBlur(p=0.2, always_apply=False),
```

Name	Type	Size	Value
ax	axes._ax...	1	Axes object of matplotlib.axes._axes module
axes	Array of object	(2, 5)	ndarray object of numpy module
BATCH_SIZE	int	1	8
best_valid_loss	float	1	inf
box	Array of float64	(4,)	[206.55999756 177.98095703 357.44000244 431.54285431]
CLASSES	list	1	['crack', 'damage', 'pothole', 'pothole_water', 'pothole_water_m']
DEVICE	device	1	device object of torch module
epoch	int	1	0
fig	figure.F...	1	Figure object of matplotlib.figure module
i	int	1	9
label	int64	1	1
model	FastRCNNPredictor	1	FastRCNNPredictor object of torchvision.models.detection.faster_rcnn module

Help Variable Explorer Plots Files

Console 1/A x

Python 3.12.3 | packaged by Anaconda, Inc. | (main, May 6 2024, 19:42:21) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.22.2 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/kahla/Desktop/Jupyter Files/Road-damage-and-pothole-detection.ipynb_Kahlawy.py', wdir='C:/Users/kahla/Desktop/Jupyter Files')
Number of training samples: 400
Number of validation samples: 100

Important

Figures are displayed in the Plots pane by default. To make them also appear inline in the console, you need to uncheck "Mute inline plotting" under the options menu of Plots.

Python Console History

conda: projEnv (Python 3.12.3) Completions: conda(projEnv) LSP: Python Line 11, Col 40 ASCII CRLF RW Mem 90%

Project Code 2

Spyder (Python 3.12)


File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\kahla\Desktop\Jupyter Files\Road-damage-and-pothole-detection.ipynb_Kahlawy.py

```
1 #####
2 # Name: Kahlawy Hussein
3 # Acadmic Number: 202301531
4 # Software Engineering Master - First Year (2023/2024)
5 # Supervised by Dr. Ashraf A. Shahin
6 #####
7
8 import numpy as np
9 import os
10 import cv2
11 from xml.etree import ElementTree as et
12 import glob
13 import torch
14 from torch.utils.data import Dataset, DataLoader
15 from torchvision.models.detection import fasterrcnn_resnet50_fpn, FasterRCNN_ResNet50_FPN_Weights
16 from torchvision.models.detection.faster_rcnn import FastRCNNPredictor
17 from torch.optim.lr_scheduler import ReduceLROnPlateau
18 from tqdm import tqdm
19 import albumentations as A
20 from albumentations.pytorch import ToTensorV2
21 import time
22 import logging
23 from mean_average_precision import MetricBuilder
24 import matplotlib.pyplot as plt
25 import matplotlib.patches as patches
26 import multiprocessing as mp
27 import warnings
28 warnings.filterwarnings("ignore")
29
30 # Constants
31 BATCH_SIZE = 8
32 RESIZE_TO = 512
33 NUM_EPOCHS = 50
34 DEVICE = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
35 TRAIN_DIR = 'C:/kaggle/input/train'
36 VALID_DIR = 'C:/kaggle/input/val'
37 CLASSES = ['crack', 'damage', 'pothole', 'pothole_water', 'pothole_water_m']
38 NUM_CLASSES = len(CLASSES) + 1 # Add 1 for background class
39 OUT_DIR = 'C:/kaggle/outputs'
40 SAVE_PLOTS_EPOCH = 5
41 SAVE_MODEL_EPOCH = 5
42
43 # Albumentations transforms
44 def get_train_transform():
45     return A.Compose([
46         A.Flip(p=0.5),
47         A.RandomRotate90(p=0.5),
48         A.MotionBlur(p=0.2, always_apply=False),
49         A.MedianBlur(blur_limit=3, p=0.1, always_apply=False),
50         A.Blur(blur_limit=3, p=0.1, always_apply=False),
51         ToTensorV2(p=1.0, always_apply=True),
52     ], bbox_params={
53         'format': 'pascal_voc',
54         'label_fields': ['labels']
```

temp.py x Road-damage-and-pothole-detection.ipynb_V1_Spider - Copy.py x Road-damage-and-pothole-detection.ipynb_Kahlawy.py x

Sample of Image Detection



Help Variable Explorer Plots Files

Console 1/A x

Python 3.12.3 | packaged by Anaconda, Inc. | (main, May 6 2024, 19:42:21) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.22.2 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/kahla/Desktop/Jupyter Files/Road-damage-and-pothole-detection.ipynb_Kahlawy.py', wdir='C:/Users/kahla/Desktop/Jupyter Files')
Number of training samples: 400
Number of validation samples: 100

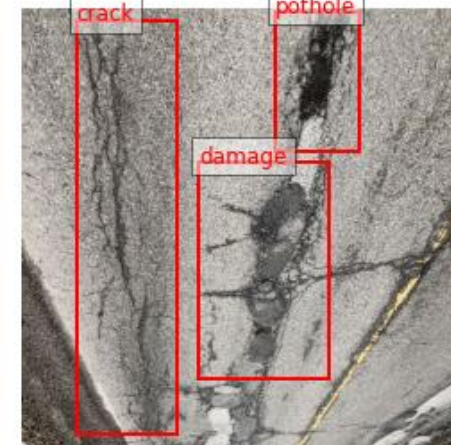
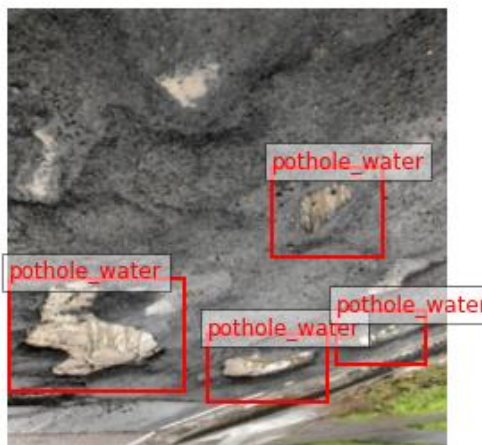
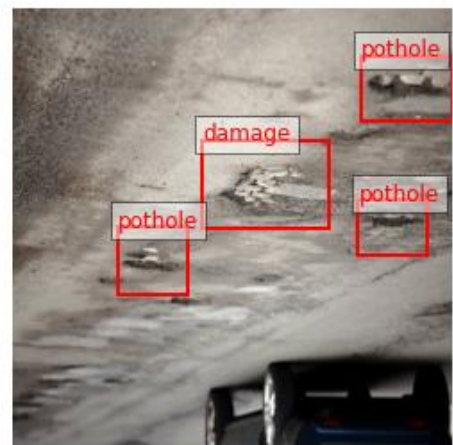
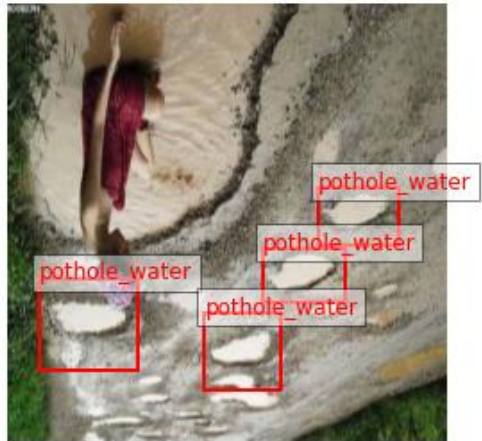
Important

Figures are displayed in the Plots pane by default. To make them also appear inline in the console, you need to uncheck "Mute inline plotting" under the options menu of Plots.

EPOCH 1 of 50
Training
0% | 0/50 [00:00<?, ?it/s]
Traceback (most recent call last):

Project Code 3

Sample of Image Detection



References

1. H. Maeda, Y. Sekimoto, T. Seto, T. Kashiya, and H. Omata, "Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127-1141, Dec. 2018.
2. Wan, F., Sun, C., He, H. et al. YOLO-LRDD: a lightweight method for road damage detection based on improved YOLOv5s. *EURASIP J. Adv. Signal Process.* 2022, 98 (2022). <https://doi.org/10.1186/s13634-022-00931-x>
3. Mraz, Alexander & Sekimoto, Yoshihide & Kashiya, Takehiro & HM, Hiroya. (2020). Development of the Localized Road Damage Detection Model Using Deep Neural Network. 1-6. 10.1145/3441233.3441235.
4. Arya, Deeksha & Maeda, Hiroya & Ghosh, Sanjay & Toshniwal, Durga & Mraz, Alexander & Kashiya, Takehiro & Sekimoto, Yoshihide. (2020). Transfer Learning-based Road Damage Detection for Multiple Countries.
5. A New Road Damage Detection Baseline with Attention Learning Article in *Applied Sciences* · July 2022 DOI: 10.3390/app12157594.
6. L. A. Silva, V. R. Q. Leithardt, V. F. L. Batista, G. Villarrubia González, and J. F. De Paz Santana, "Automated Road Damage Detection Using UAV Images and Deep Learning Techniques," in *IEEE Access*, vol. 11, pp. 62918-62931, 2023, doi: 10.1109/ACCESS.2023.3287770.
7. Saha, P. K., Arya, D., & Sekimoto, Y. (n.d.). Federated learning-based global road damage detection. *Computer-Aided Civil and Infrastructure Engineering*. Retrieved from <https://doi.org/10.1111/mice.13186>.
8. Mustakim, Md Mizanur. (2023). Road Damage Detection Based on Deep Learning. 10.13140/RG.2.2.22523.28967.
9. Chen, Xinhao & Liu, Chang & Chen, Long & Zhu, Xiaodong & Zhang, Yaohui & Wang, Chenxi. (2024). A Pavement Crack Detection and Evaluation Framework for a UAV Inspection System Based on Deep Learning. *Applied Sciences*. 14. 1157. 10.3390/app14031157.
10. Merkle, Nina & Henry, Corentin & Azimi, S. & Kurz, Franz. (2022). ROAD CONDITION ASSESSMENT FROM AERIAL IMAGERY USING DEEP LEARNING. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. V-2-2022. 283-289. 10.5194/isprs-annals-V-2-2022-283-2022.
11. Rodriguez-Lozano, F. J., Gámez-Granados, J. C., Palomares, J. M., & Olivares, J. (2023), Efficient data dimensionality reduction method for improving road crack classification algorithms. *Computer-Aided Civil and Infrastructure Engineering*, 38, 2339–2354. <https://doi.org/10.1111/mice.13014>
12. Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Omata, H., Kashiya, T., & Sekimoto, Y. (2022). Crowdsensing-based Road Damage Detection Challenge (CRDDC-2022). arXiv preprint arXiv:2211.11362.
13. Pan, Y., Zhang, X., Cervone, G., & Yang, L. (2018). Detection of asphalt pavement potholes and cracks based on the unmanned aerial vehicle multispectral imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(10), 3701-3712. <https://doi.org/10.1109/JSTARS.2018.2865528>
14. Majidifard, H., Adu-Gyamfi, Y., & Buttlar, W. G. (2020). Deep machine learning approach to develop a new asphalt pavement condition index. *Construction and Building Materials*, 247, 118513. <https://doi.org/10.1016/j.conbuildmat.2020.118513>
15. Chen J, Yu X, Li Q, et al. LAG-YOLO: Efficient Road damage detector via lightweight attention ghost module. *Journal of Intelligent Construction*, 2024, 2(1): 9180032. <https://doi.org/10.26599/JIC.2023.918003>
16. H. Liang, S.-C. Lee, and S. Seo, "Automatic Recognition of Road Damage Based on Lightweight Attentional Convolutional Neural Network," *Sensors*, vol. 22, no. 24, p. 9599, 2022. [Online]. Available: <https://doi.org/10.3390/s22249599>.
17. Arya, Deeksha; Maeda, Hiroya; Ghosh, Sanjay Kumar; Toshniwal, Durga ; Omata, Hiroshi ; Kashiya, Takehiro ; Seto, Toshikazu; Mraz, Alexander ; Sekimoto, Yoshihide (2021), "RDD2020: An Image Dataset for Smartphone-based Road Damage Detection and Classification", Mendeley Data, V1, doi: [10.17632/5ty2wb6gvg.1](https://doi.org/10.17632/5ty2wb6gvg.1)

Dataset and Python Code

- Dataset URL:
<https://www.kaggle.com/datasets/trolololo888/potholes-and-road-damage-with-annotations>
- Source Code:
<https://github.com/kahlawy/RoadDamageDetection>
- Updated Research Paper:
[RoadDamageDetection/Research_Paper](#) at main · kahlawy/RoadDamageDetection · GitHub